Amendments to the Specification:

Please replace paragraph [0098] with the following amended paragraph:

[0098] Figure 9 is a flow diagram of a procedure for generating the 22 hash indices for "distance two points." That is, an attribute vector defining points a Hamming distance to and from the center of a decoding sphere, when subject to additional one bit distortions of the 23 bit code words, result in code words found in a total of 22 code words spheres, the originating or central index sphere and 21 adjacent code spheres. Thus, Step 901, the process receives a 23-bit "weight 2" attribute vector and, at Step 902 an arbitrary unit vector is added to the 23-bit vector to create a new vector of weight 3. At Step 903, the newly created weight 3 vector is used to generate six hash keys using part of the procedure shown in detail in connection with Figure 8. Thus, the output provided at Step 806 (Figure 8) is output at Step 904 including the six hash keys, *i.e.*, one central and five peripheral keys. At Step 905 two of the peripheral indices V and W are arbitrarily selected and four unit vectors (V_1 V_2 V_3 V_4 and W_1 W_2 W_3 W_4) corresponding to each are identified. At Step 906 16 weight 2 vectors U_k are created by pairwise summing combinations of the unit vectors V_I and W_j (*i.e.*, the center and off set values) identified in Step 905. At Step 907 16 weight 4 vectors S_k are created from the U_k vectors by adding each to the original 23-bit input vector. A hash transform is applied to Step ~~808~~ 908 to each of the resultant S_k vectors to generate 16 hash keys and, together with the previously generated 6 hash keys output at Step 904, are output for a total of 22 hash keys at Step 909.

Please replace paragraph [0109] with the following amended paragraph:

[0109] Another method of identifying data items satisfying a particular maximum distortion criteria (*e.g.*, within a distance 2 of a target data item) and/or returning a list without duplicates relies on a pairing table which is initialized to all zeros when the data structures are loaded. Referring to Figure 14, extracted data items using the pairing table are provided as lists from hash keys at step 1401. At step 1402, the table entry corresponding to the data item is incremented. That is, if q is a data element in the list, then the next record is then indexed, *i.e.*, PairingTable [q] = PairingTable [q] + 1. At step ~~1402~~ 1403, the data item "q" is output if the count contained in PairingTable [q] = 2, *i.e.*, there is at least one match of the hash keys. A test is performed at 1404 to see if there are more elements in the list to process and, if so, branch back via step 1405 to step 1402. Conversely, if the list is exhausted, then processing continues via step 1406 to step 1407 which selectively, for each element q, sets Pairing Table [q] = 0. It should be noted that this last step selectively reinitializes the Pairing Table so as to reset only those records requiring resetting instead of arbitrarily reinitializing all values to 0. One skilled in the art would recognize that there is a substantial resource and time savings realized by selective reinitialization of only those entries affected by the process.